BAT and RDT Setup Guide

Based on Monica C2080 PV

Hugo SF Tsai

2021/10/05

wiwynn

Content

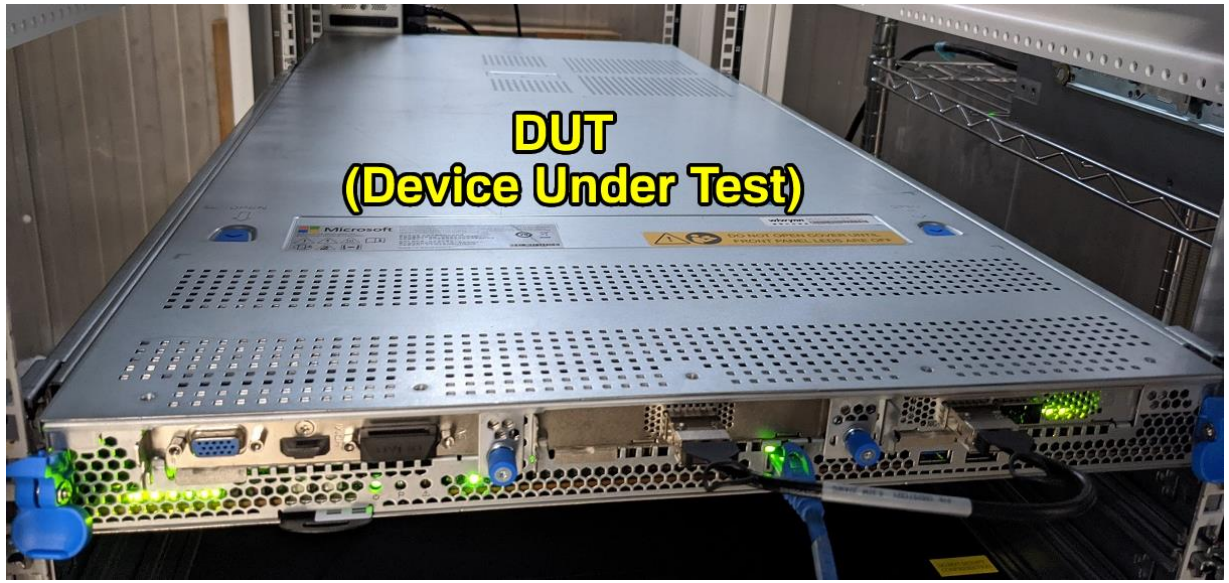
- **1) What to Know First**
  - Understanding common but important terms before we start
- **2) All You Need Before Test**
  - Checking some items in advance to avoid any delays
- **3) Scripts Used in Test**
  - Explaining what and how the script works
- **4) Executing the BAT and RDT**
  - Learning how to execute the BAT and RDT
- **5) Understanding the Test Result**
  - Reading and checking the results after a test

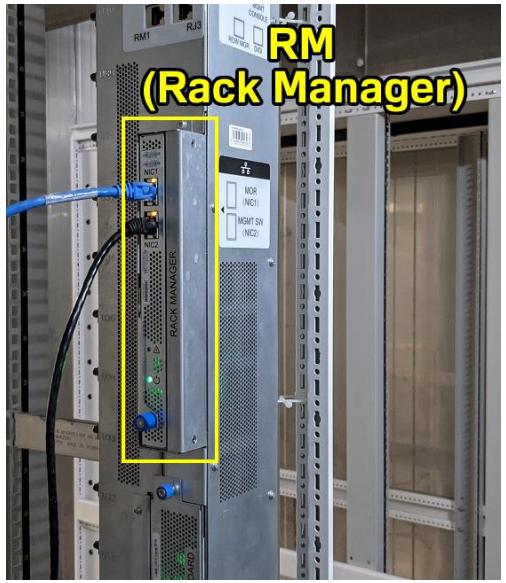
What to Know First

Understanding common but important terms before we start

What to Know First

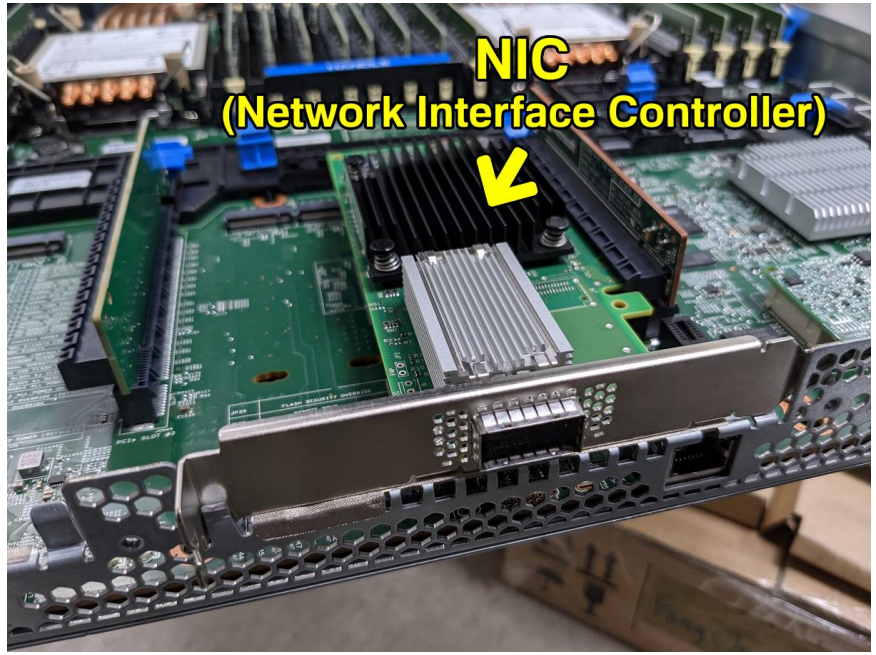
- **Before we start to setup an environment for reliability test, there are few terms we need to know first. These terms appears in this document several times, so it is very important and necessary to understand it correctly:**
  - **DUT** (Device Under Test) is a manufactured product undergoing the reliability testing
  - **RM** (Rack Manager) is an interface which can control rack and report measurement

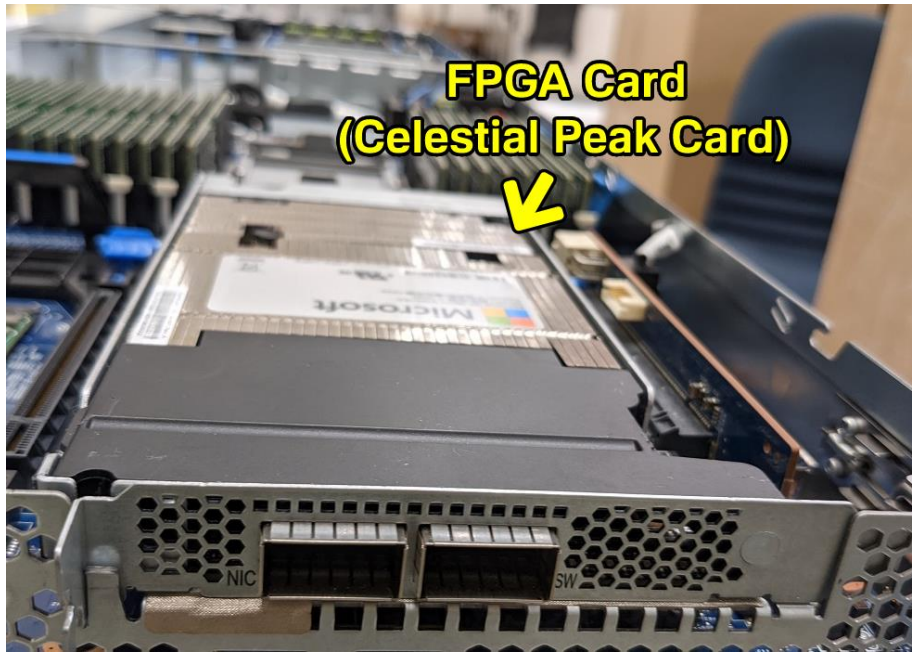




What to Know First

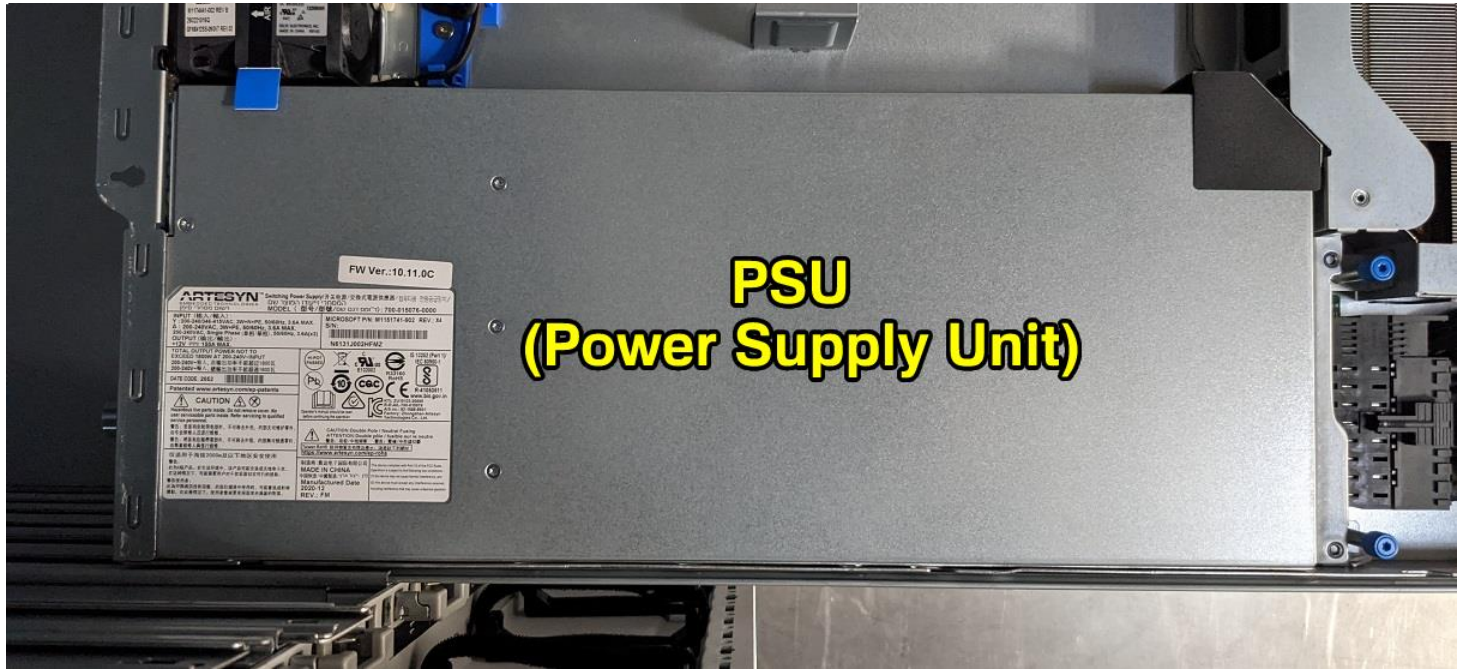
- **Before we start to setup an environment for reliability test, there are few terms we need to know first. These terms appears in this document several times, so it is very important and necessary to understand it correctly:**
  - **NIC** (Network Interface Controller) is a hardware that connects computer to network
  - **FPGA** (Field Programmable Gate Array) is an IC designed to be configured by customer

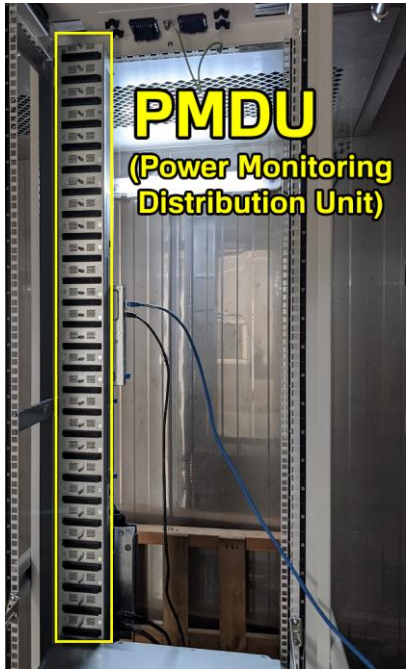




What to Know First

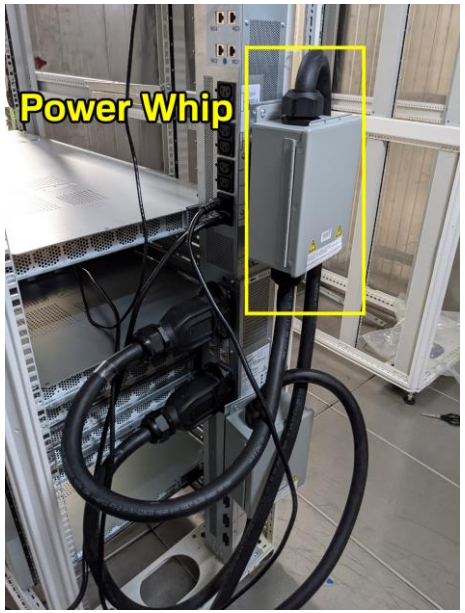
- **Before we start to setup an environment for reliability test, there are few terms we need to know first. These terms appears in this document several times, so it is very important and necessary to understand it correctly:**
  - **PSU** (Power Supply Unit) is a unit which convert AC power to DC power for internal component
  - **PMDU** (Power Monitoring Distribution Unit) is a unit provides power and signal for each DUT

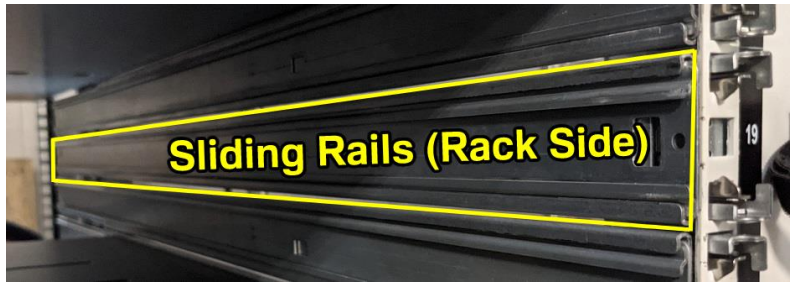




What to Know First

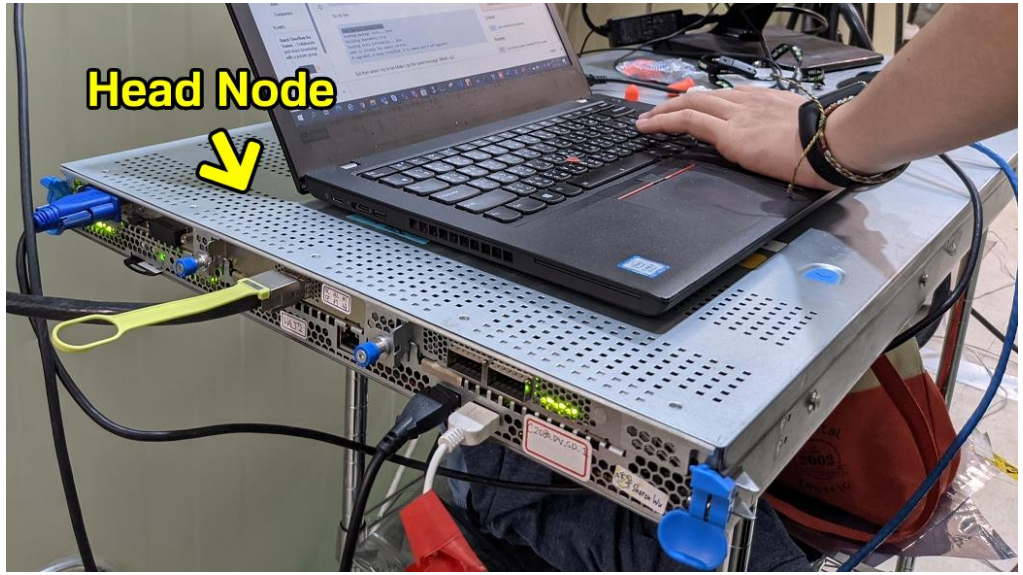
- **Before we start to setup an environment for reliability test, there are few terms we need to know first. These terms appears in this document several times, so it is very important and necessary to understand it correctly:**
  - **Rack** is a chassis that holds DUTs, Switch and other must-used hardware
  - **Power Whip** is a fuse box that connects power source to the PMDU
  - **Sliding Rails** are metal pieces installed at both rack and DUTs for holding DUTs on rack

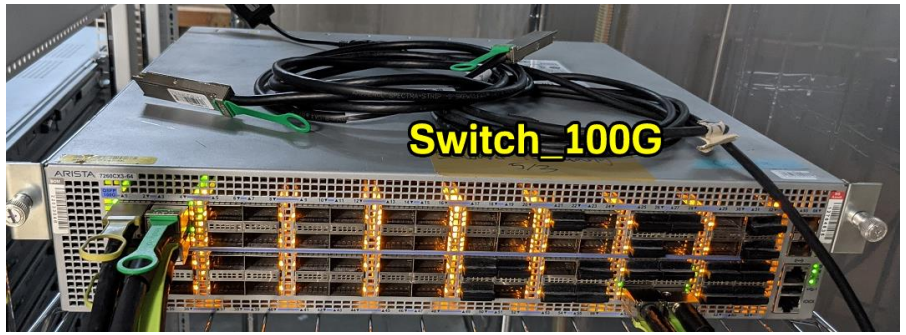




What to Know First

- **Before we start to setup an environment for reliability test, there are few terms we need to know first. These terms appears in this document several times, so it is very important and necessary to understand it correctly:**
  - **Head Node** is a device managed by engineer which can monitor and control DUTs
  - **Switch** is a networking hardware that connect devices on a computer network





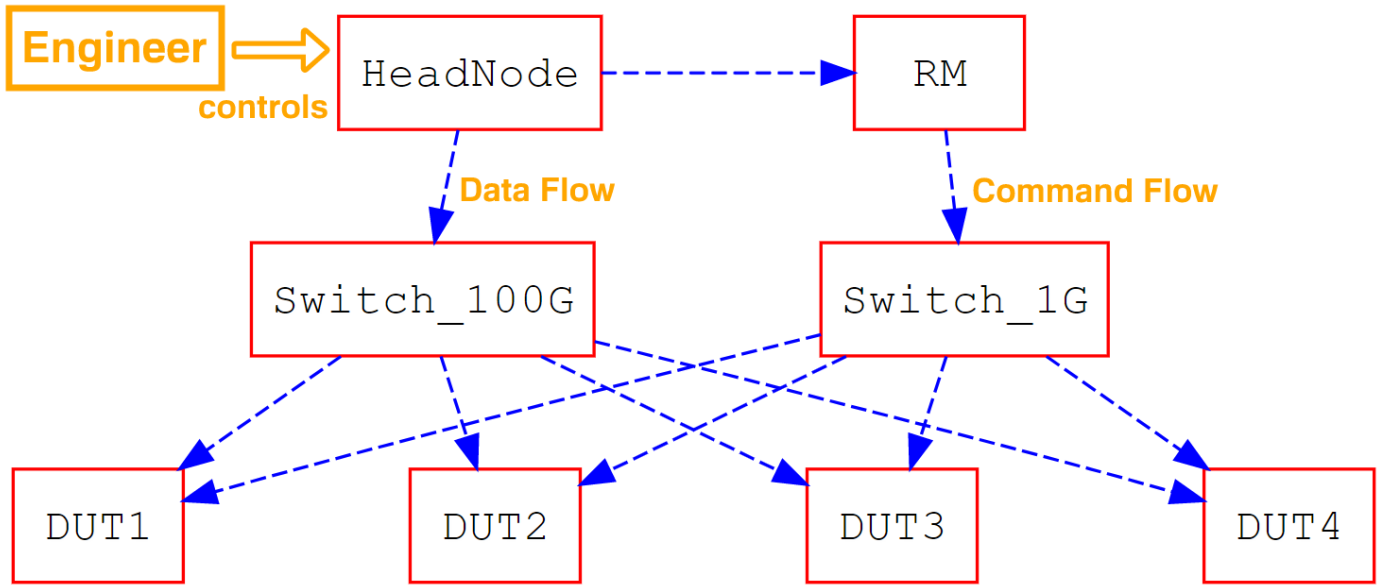
What to Know First

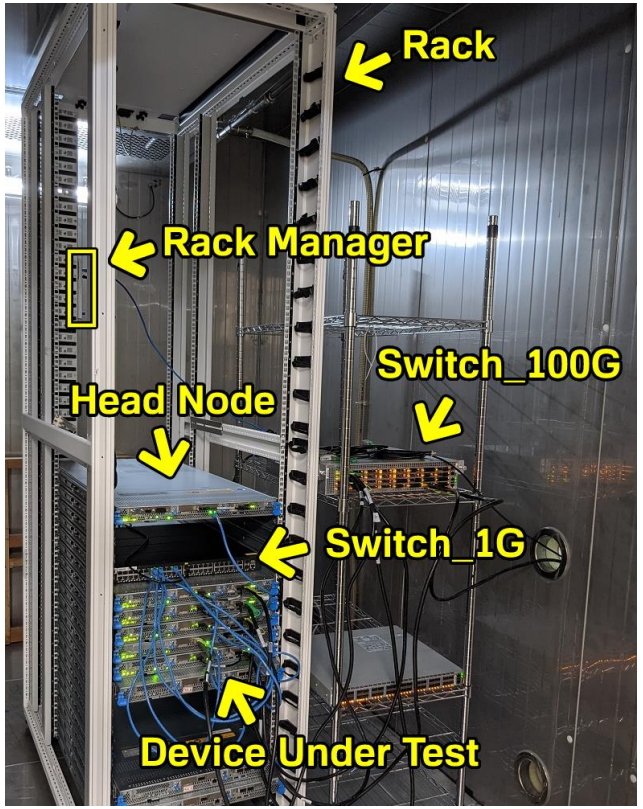
- **Before we start to setup an environment for reliability test, there are few terms we need to know first. These terms appears in this document several times, so it is very important and necessary to understand it correctly:**
  - **BAT** (Baseline Acceptance Test) is a test to determine whether the system meets the requirement
  - **RDT** (Reliability Demonstration Test) is a process of demonstrating the reliability of a product
  - BAT/RDT are the main tasks to achieve for reliability test. This document focus on how to setup DUTs for BAT/RDT, how to execute BAT/RDT and how to collect logs from DUTs

  - **OS** (Operation System) is a computer system software that manages the hardware and software
  - **PXE** (Preboot eXecution Environment) is a client-sever environment that boots from a network
  - **BIOS** (Basic Input/Output System) is a standard firmware used to perform hardware initialization
  - **BMC** (Baseboard Management Controller) is a small computer on motherboard controlled by IPMI
  - **IPMI** (Intelligent Platform Management Interface) is a hardware management interface with BMC
  - Above items are basically software and firmware, there are no physical hardware

What to Know First

- **The rack which is ready for test can be referred to the following picture:**





Example

- Connecting NIC1 port on RM to 1G LAN port on Head Node
- Connecting NIC2 port on RM to port 25 (RM port) on Switch_1G
- Connecting 1G port on Switch_1G to corresponded 1G LAN port on DUTs
- Connecting 100G port on Switch_100G to 100G LAN port on DUTs

All You Need Before Test

Checking some items in advance to avoid any delays

All You Need Before Test

- **Document: Test Plan**
  - A test plan is for engineers to prepare scripts, firmware, chamber and other stuff in advance

C2080 Test case_RDT 47°C/80%RH

Category	Test Objective	Test Level	Test Criteria	Test condition
Operation	Put the system in environment Temperature/Humidity 43.5° C/80%RH for 1008 hrs(Make sure system inlet temp is around 47oC±2°C during running system stress.). It's OP test case that included AC power cycling & system stress. Run stress after it power on & boot into OS for 150 mines for each power cycle.	System	1. Power ON and functional test without failure. 2. No component damage. 3. Visual inspection for components corrosion and delamination. No defect is allowed	Temperature : 47°C(sys_inlet) Humidity : 80%

* Workflow for DMTBF test.

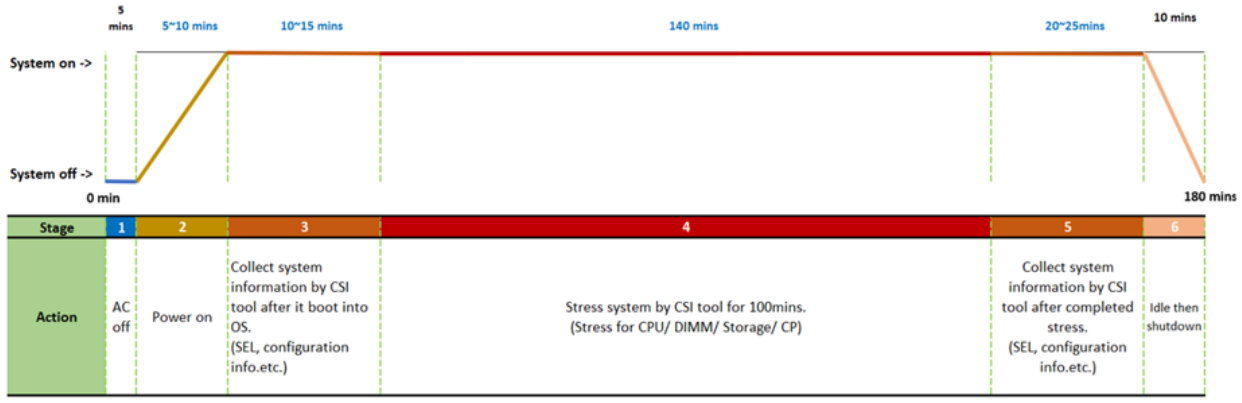
Full stress 180 mins then system DC off, after DC off use RACK PMDU to AC off & AC on systems.

1. System boot to Windows OS then execute full stress patch data.

2. After stress 140 mins, system will collect system relevant data then auto DC off self.

3. Then use RACK PMDU to AC off/on 1 time for all of the system.

4. Loop step1~3 1008hrs



DMTBF Test condition		
Full stress	Test Schedule	2021/05/1 - 2021/06/18 *(1008hrs)*
	Working Days	49 days
	Sample size	20 pcs

All You Need Before Test

- **Document: Cross Table**
  - A cross table is to confirm we use the DUT with the correct configuration for test

C2080 PV Gen8.2 L10 Project Code: BPD033010001								1U	1U	1U	1U	1U	1U-QUANTA
Level	TYPE				Manufacture		Remark	L10 SERVER GP-MM GEN8.2	L10 SERVER GP-LM-GEN8.2	Loose L10 w/packing L10 SERVER GP-MM GEN8.2 W/PACKING	Loose L10 w/packing L10 SERVER GP-LM-GEN8.2 W/PACKING	L10 SERVER GP-MM GEN8.1	GEN8 L10 SERVER 1U-QUAN
							Site	F912	F912	F912	F912	F912	F136
							L10 MS PN						
							L10 WIW PN	B81.03301.0068	B81.03301.0069	B81.03301.0072	B81.03301.0073	M1163581-001$EK03	B81.03301.0054
							Remark						
							Qty per rack/ Loose L10 qty	15	16	62	38	15	
Level	TYPE				Manufacture		Remark						
Board													
1	1U FPGA Riser	M1126123-001	M1126123-001$A02	GEN8 PV 1U FPGA RISER	WISTRON			1	1	1	1	1	1
A	1U FPGA Riser	M1126123-001	M1126123-001$A	GEN8 PV 1U FPGA RISER	WISTRON			A	A	A	A	A	
1	1U PCIE Riser	M1126119-001	M1126119-001$A02	GEN8 PV 1U PCIE RISER	WISTRON			2	2	2	2	2	2
A	1U PCIE Riser	M1126119-001	M1126119-001$A	GEN8 PV 1U PCIE RISER	WISTRON			A	A	A	A	A	
L6 BZA													
1	Gen8 L6-AVC Fan-WZS	M1129970-003	M1129970-003$B	GEN8 L6,P2011 PSU_DELTA,FAN_AVC,WZS	WISTRON								1
1	GEN8.2 L6-DELTA Fan	M1129970-003	M1129970-003$BK00	GEN8.2 L6,P2011 PSU_DELTA,FAN_DEL	WISTRON			1					
1	Gen8.2 L6-AVC Fan	M1129970-003	M1129970-003$BK01	GEN8.2 L6,P2011 PSU_DELTA,FAN_AVC	WISTRON				1	1	1		
1	GEN8.1 L6-DELTA Fan		M1129970-001$EK00	GEN8.1 L6,P2011 PSU_DELTA,FAN_DEL	WISTRON							1	
ME Parts (L10)													
1	CPU carrier	M1124087-001	022.70063.0001	CARRIER ICE LAKE 2-2330552-2	TE			2	2	2	2	2	2
CPU REMOTE HEAT SINK													
1	Remote HS main AVC	M1140201-001	B34.03303.0002	HSINK_1U REMOTE_C2080_AVC	AVC			1	1	1	1	1	1
CPU													
1	XCC PRQ 99A9F3	M1172617-001	001.00ICX.M001	99A9F3 CD8068904586906 RKHC IC CPU INTEL ICX 8370C 32C 2.8G LGA	Intel			2	2	2	2	2	2
A	XCC PRQ 99A9F3	M1172617-001	BCS.03301.0008	IC CPU INTEL ICX 8370C 32C 2.8G LGA	Intel					A	A	A->0	
RDIMM													
1	Hynix 32G	M1171184-001	M1171184-001$A	MMOD,32GB,SM,DDR4,HYNIX,HMA84GR7DJR4~~	SK HYNIX	HMA84GR7DJR4N-XN		16	16	16	16	16	
1	Hynix 16G	M1171185-001	M1171185-001$A	MMOD,16GB,SM,DDR4,HYNIX,HMA82GR7DJR4~~	SK HYNIX	HMA82GR7DJR4N-XN		16		16		16	
1	Micron 32G	M1152909-001	M1152909-001$A	MMOD,32GB,DDR4,MICRON,MTA18ASF4G72PZ-3G2	Micron								16
Add-On													
1	Celestial Peak -PV	M1096519-005	M1096519-005$B	CELESTIAL PEAK A2040 FPGA CARD PV2.5	QUANTA			1	1	1	1	1	1
1	CX-5	M1146245-001	M1146245-001$B	NIC MELLANOX MCX515A-CCAT 100G SINGLE	Mellanox			1	1	1	1	1	

All You Need Before Test

- **Document: Firmware Recipe**
  - A firmware recipe is to check firmware and driver version meets requirement

Stage:	PV	Version :	R6	Updated :	8-Apr-21	
Group	**Item**	**Model**	**MPN**	**FW/OS Version**	**Remark**	Wiwynn Download Path
Compute Blade	MB BIOS	C2080		C2080.BS.1C16.GN5 with 0x280 microcode hot patch	Config A, A1, B, B1, D, D1 :C2080.BS.1C16.**AE1** Config C, C1 (ACC) : C2080.BS.1C16.**ZC1**	
	Chipset	Chipset driver	BKC WW11/WW14	Chipset- 10.1.18661.8255		https://pts.wistron.com/~pts/subsystem/dts/project_view.php?file=1617027220007084
	BMC	C2080		C2080.BC.0305.00	PV MB	https://pts.wistron.com/~pts/subsystem/dts/project_get_file.php?file=1617084390469410
	Fan table	refer to service config sheet				
	SDR	refer to service config sheet				
	Inventory	refer to service config sheet				
	Cerberus	FW	M1165085-001$A2	1.5.2.1	by pass mode -> Active mode except for servers that require test loads.	https://pts.wistron.com/~pts/subsystem/dts/project_view.php?file=1617845170086218
	Cerberus	PCD		C2080-Gen8.2.PCD.0x1.bin	For cerberus active mode with seamless update enabled	https://pts.wistron.com/~pts/subsystem/dts/project_view.php?file=1617845317946155
	Cerberus	Utility	M1165085-001	1.4.8.1		https://pts.wistron.com/~pts/subsystem/dts/project_view.php?file=1617027338212508
	TPM	C2030	M1039577-001	7.85.4555.0		
	CPLD	checksum: 02745BD3		C2080_V05		
	SSD M.2	Hynix PE6110 /960G	HFS960GD0FEI-A430A	11030G00_1T_signed_mp		https://pts.wistron.com/~pts/subsystem/dts/project_view.php?file=1611297441702181
	SSD M.2	Hynix PE6110 /1.92TB	HFS1T9GD0FEI-A430A	11030G00_2T_signed_mp		https://pts.wistron.com/~pts/subsystem/dts/project_view.php?file=1611297441702181
	SSD M.2	Samsung PM9A3 / 960G	MZ1L2960HCJR-000AMV	GDC73M4Q	support OCP1.0a+FIPS+ECO	https://pts.wistron.com/~pts/subsystem/dts/project_view.php?file=1617780062187344
	SSD M.2	Samsung PM9A3 / 1.92TB	MZ1L21T9HCLS-00AMV	GDC73M4Q		https://pts.wistron.com/~pts/subsystem/dts/project_view.php?file=1617780062187344
	Mellanox CX-5	Mellanox NIC card - FW Rev	M1146245-001 MCX515A-CCAT_C11	16.25.8368	Azure-20201108.mfa	https://pts.wistron.com/~pts/subsystem/dts/project_view.php?file=1604958253621916
		Mellanox NIC card - uEFI Rev		14.18.23		
		Mellanox NIC card - PXE Rev		3.5.702		
	FPGA	Celestial Peak	M1096519-002	RevH.Gamma	CELESTIAL PEAK,A2040, Rev H.Gamma (Drop22.1)	https://teams.microsoft.com/_#/files/General?threadId=19%3A139bfb0533144e1a800079f
		Golden Image		Shell Package: 3.13.14 Role ID: 0xcaf1601d	CelestialPeak_SysInt_3.13.14-2641ca46_jic.rpd	update FPGA impage through Intel Quartus programmer https://cloudadmwiwynn.sharepoint.com/teams/C2080Gen8/Shared%20Documents/General/Driver
		Factory Tester Image		Shell Package: 3.13.14 Role ID: 0xcaf20fac	CelestialPeak_SysInt_3.13.14-2641ca46_jic,rpd	
		Host FPGA drivers		5.16.1.4		
		SoC FPGA SW		5.16.1.4		
		SoC OS		1908.5.14981755	drop22.1.dev-stos13 Linux localhost 5.4.83-microsoft-standard #1 SMP Sun Feb 7 07:02:18 UTC 2021 aarch64 GNU/Linux	

All You Need Before Test

- **Hardware:**
  - Rack: we need to know how many DUTs will be used and prepare enough racks
  - Sliding Rails: same as above, knowing the amount will help the tester to prepare rails
  - PMDU: a power management unit for power up the rack, a must-have component in every rack
  - Power Whip: for connecting PMDU to the power sources
  - Power Cord: for Head Node and Switches (it might have different types of power cords)
  - RJ45 Cables: for connecting 1G Switch to DUTs, Head Node and RM
  - QSFP Cables : for connecting 100G Switch to DUTs and Head Node
  - Head Node: for controlling and monitoring DUTs (better to prepare more than 1 Head Node)
  - DHCP Server: for sending IP to Celestial Peak (it will enable the NIC on DUT)
  - PXE Server: for installing required OS to all DUTs (automatically install is much faster)
  - Rack Manager: for managing the rack, another must-have component in every rack
  - 1G Switch: for building connections between RM and DUTs
  - 100G Switch: for building connections between Head Node (or PXE server) and DUTs

All You Need Before Test

- **Software & Firmware:**
  - OS, Chipset Driver and NIC Driver
    - These items are the easier ones in this types, just need to check the version meets the requirement
    - Chipset Driver and NIC Driver can be packed in a single OS then use PXE server to deploy
  - Firmware for BIOS, BMC, TPM, NIC, FPGA, PSU, RM, Cerberus, Celestial Peak and Celestial Cerberus
    - Above firmware need to flash one at a time, but they have different ways to flash
    - Although firmware need to flash one by one, we can use script to flash it the whole rack through RM
    - Not all firmware need to flash, we need to check firmware recipe for correct versions
    - If any error occurred in the flashing process, contact project leader or R&D for help
  - Script
    - Because we have so many things to do before and after tests, we need script to speed up the process
    - We might use different scripts based on different project or even different SKUs
    - We use windows batch type script in C2080 due to the OS (Windows Environment)
    - The next chapter will introduce some of script that we used in C2080 RDT

Scripts Used in Test

Explaining what and how the script works

Scripts Used in Test

- **All used scripts can be divided into 2 types based on usages at different phase. The first one is SETUP related scripts:**
  - Rename your DUTs
    - We need to rename all DUTs based on different SKUs, so that we can recognize it from remote.
  - Connect all net disks to head node
    - We need to connect all DUTs to head node for controlling and monitoring
  - Copy and Paste required files
    - We need to copy the files we need from A to B (maybe do it automatically at every specific time)
  - Mount and Format Disks
    - We need to mount and format disks to DUTs for stress test
  - Flash Firmware
    - We need to flash firmware to each commodities with different ways (such as commands and scripts)

Scripts Used in Test

- **The second type is TEST related scripts, we usually use these script to execute stress test, dump logs and report error messages automatically:**
  - Execute Stress Test
    - We need to execute stress tool automatically (prime95 and io-meter)
  - Dump Logs
    - We need to collect logs before, after and during the test, so that we can know the test status
    - SDR (Sensor Data Reading) for monitoring temperatures, power, voltage and fan speed
    - SEL (System Event Log) for collecting unusual or error messages from system
  - Sort Result and Report Error Messages
    - We need to sort and check logs then report test result to engineer

Scripts Used in Test

- **Here is a script example for renaming all DUTs:**

```
@echo off
SetLocal ENABLEDELAYEDEXPANSION
cd ..
chdir > path.txt
FOR /F "delims=" %%c IN ('type path.txt') DO SET PA=%%c
echo !PA!

cd !PA!\2_Rename
set /a i=1
FOR /F  %%k IN ( !PA!\Information\Name.txt) DO (SET sys!i!=%%k
set /a i+=1
)

echo !PA!

set /p s=enter system qty

for /l %%y in ( 1,1,%s% ) do (
echo @echo off >> !sys%%y!_rename.bat
echo wmic ComputerSystem Where Name="%%ComputerName%%" Rename Name=!sys%%y! >> !sys%%y!_rename.bat
echo wmic ComputerSystem Where Name="%%ComputerName%%" Rename Name=!sys%%y!

)
echo end
timeout /T 30 /nobreak > nul
```

Scripts Used in Test

- **Here is another script example for connecting all net disk to head node:**

```
@echo off
SetLocal ENABLEDELAYEDEXPANSION
cd ..
chdir > path.txt
FOR /F "delims=" %%c IN ('type path.txt') DO SET PA=%%c
echo !PA!

cd !PA!\3_Netuse
set /a i=1
FOR /F  %%k IN ( !PA!\Information\Name.txt) DO (SET sys!i!=%%k
set /a i+=1
)

set /a j=1
FOR /F  %%p IN ( !PA!\Information\Disk.txt) DO (SET ndisk!j!=%%p
set /a j+=1
)

set /p s=enter system qty
for /l %%y in ( 1,1,%s% ) do (
net use !ndisk%%y!: \\!sys%%y!\C$ /user:administrator p@ssw0rd
echo net use !ndisk%%y!: \\!sys%%y!\C$ /user:administrator p@ssw0rd
)
timeout /T 30 /nobreak > nul
```

Executing the BAT and RDT

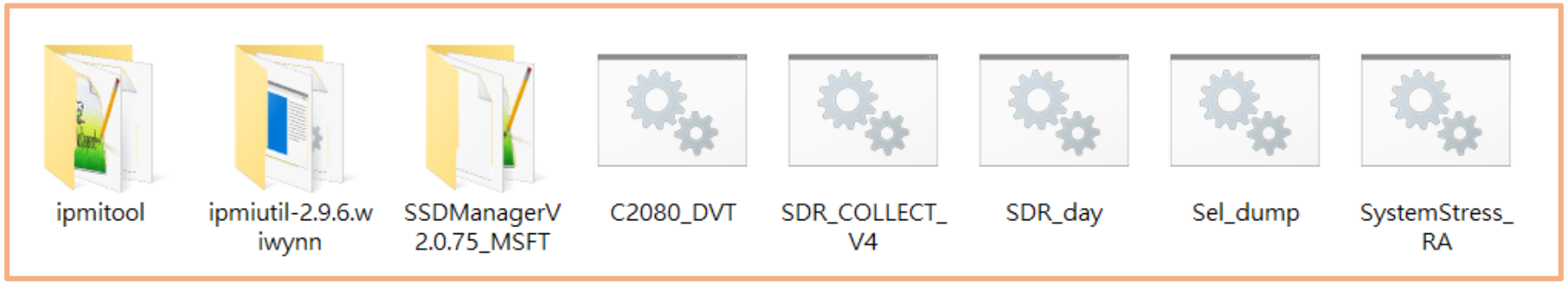
Learning how to execute the BAT and RDT

Executing the BAT and RDT

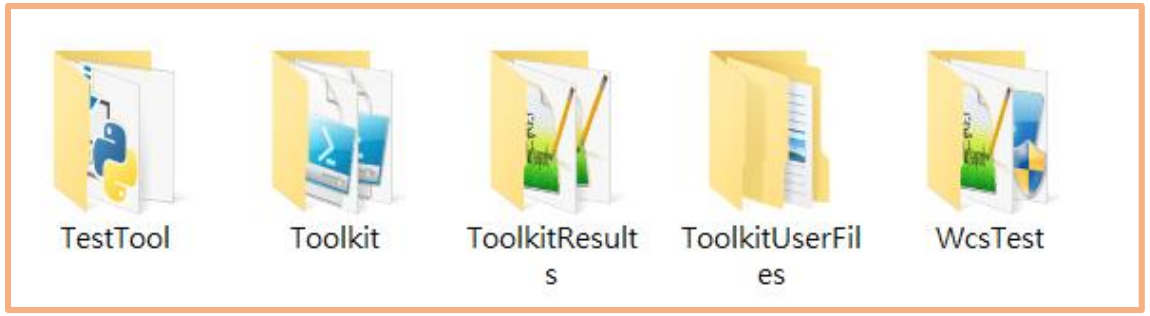
- **What to do in both BAT/RDT?**
  - Stress Test - Both BAT and RDT will give stress on CPU, RAM and Storage
  - Log Collecting - Both BAT and RDT will collect logs from stress tool and IPMI sensors during tests
    - Test Result - Checking the result is pass or fail at logs from CPU, RAM and Storage tests
    - IPMI SEL (System Event Log) - Checking the system event from IPMI event recorder
    - IPMI SDR (Sensor Data Record) - Checking the sensor data from IPMI sensor recorder
    - Windows Event Viewer - Checking the abnormal event from Windows event viewer
- **What is the difference between BAT and RDT?**
  - Test Duration
    - BAT is running for 12 hours and only do it for single cycle
    - RDT is running for 4 weeks (or more); about 3-4 hours each cycle
  - Test Method
    - BAT is a continuous test, no need to reboot during it
    - RDT need to reboot between each cycle (Power on and off by RM)

Executing the BAT and RDT

- **Copy and Paste the Files**
  - Before executing the test, we need to paste some files to specific path
  - Some of these files were provided by Microsoft and some of it were created by our side
    - Paste the following folders and files to **C:\Users\Administrator\Desktop**



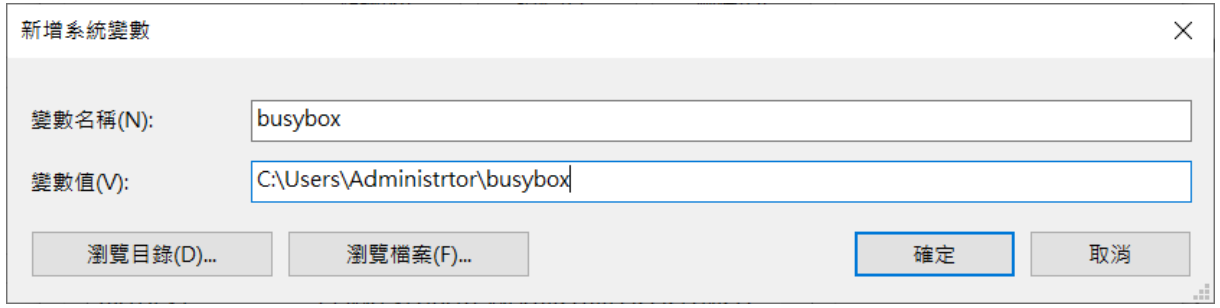
    - And paste these folders and files to **C:**



Executing the BAT and RDT

- **Install busybox**
  - Because windows environment does not support Linux command, we need to install busybox for accomplishing the function write in Linux command in test scripts.
  - Step by Step installation procedure as below
    - Step1. Download busybox from https://frippery.org/busybox/

      From time to time binary builds and source tarballs will be made available. The latest version is always busybox.exe (currently this is an alias for busybox-w32-FRP-4264-gc79f13025.exe). Release notes for this version are available.
    - Step2. Place the busybox.exe to path C:\Users\administrator\busybox (create manually if it does not exist)
    - Step3. Open a cmd windows in this path and type busybox --install
    - Step4. Go to Control Panel > System > Advanced System Settings > Advanced > Environment Variables
    - Step5. Add busybox to PATH in **both** user variable and system variable section



Executing the BAT and RDT

- **Modify the Test Duration**
  - In BAT, we can modify test duration in file **2080_DV.bat** (line 39)

```
36  ::  START CP_stress.bat
37  rem Quickstress
38      CD "C:\WcsTest"
39      PowerShell -Command ". ..\WcsTest\Scripts\wcsScripts.ps1;Run-Quickstress -TimeinMin 20 -Nonet -Nofpga
40  rem Post-test
41      PowerShell -Command ". ..\WcsTest\Scripts\wcsScripts.ps1;Post-WCSTest -ResultsDirectory C2080_DV_BASELINE
42
```

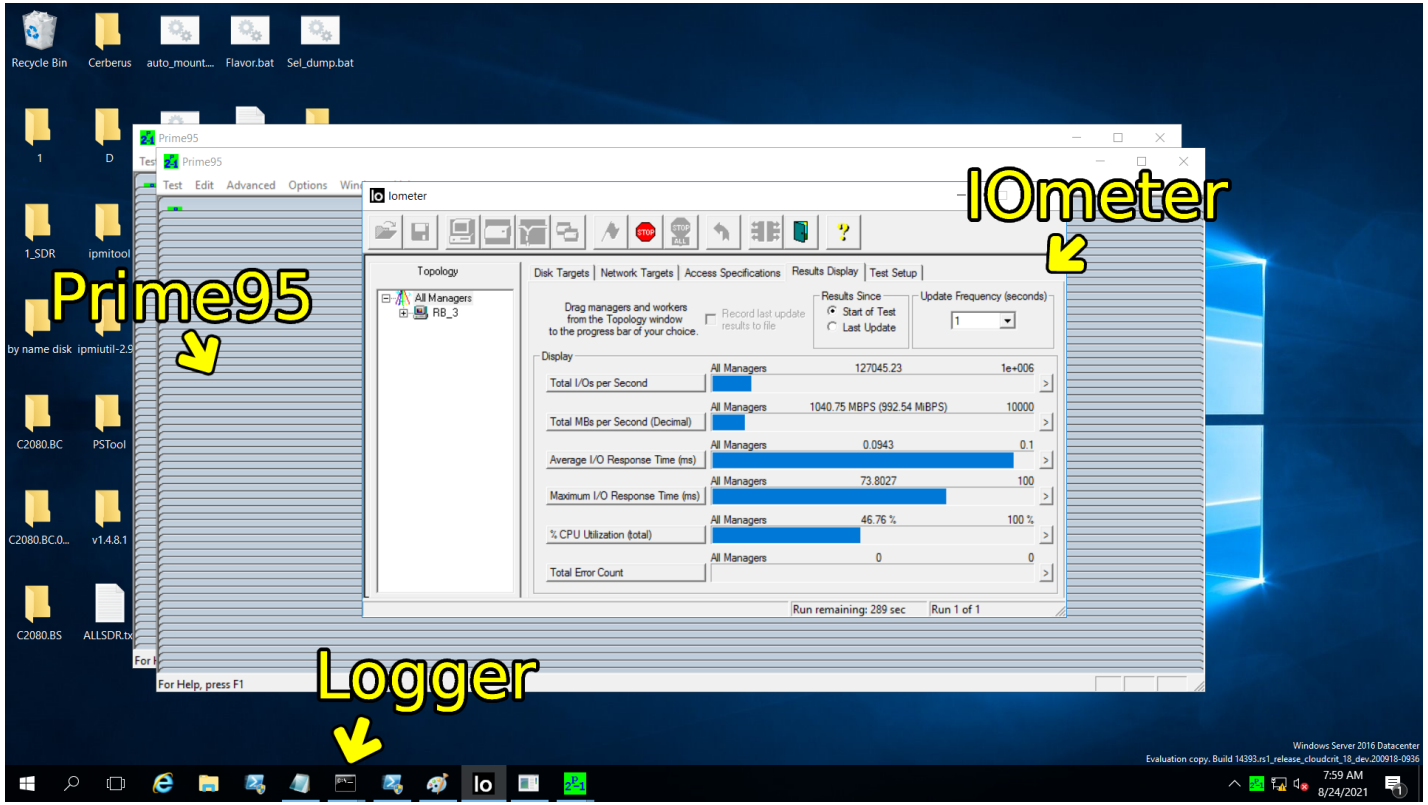
  - In RDT, we can modify test duration in file **SystemStress_RA.bat** (line 50)

```
44  rem Sel_20min
45      CD "%desk%"
46      start /min Sel_dump.bat
47  rem Main Test
48      CD %csi%
49  rem pretest+Quickstress+posttest
50      PowerShell -Command ". ..\Toolkit\StartToolkit.ps1;Run-Quickstress -NoNetwork -TimeinMin 180
51
```

  - Time unit in both files are minute

Executing the BAT and RDT

- **Execute the Trial Run**
  - After we modified time duration, we had better do a trial run before formal test
  - Make sure **logger** and **stress tools** (Prime95 and IOmeter) are running correctly



Understanding the Test Result

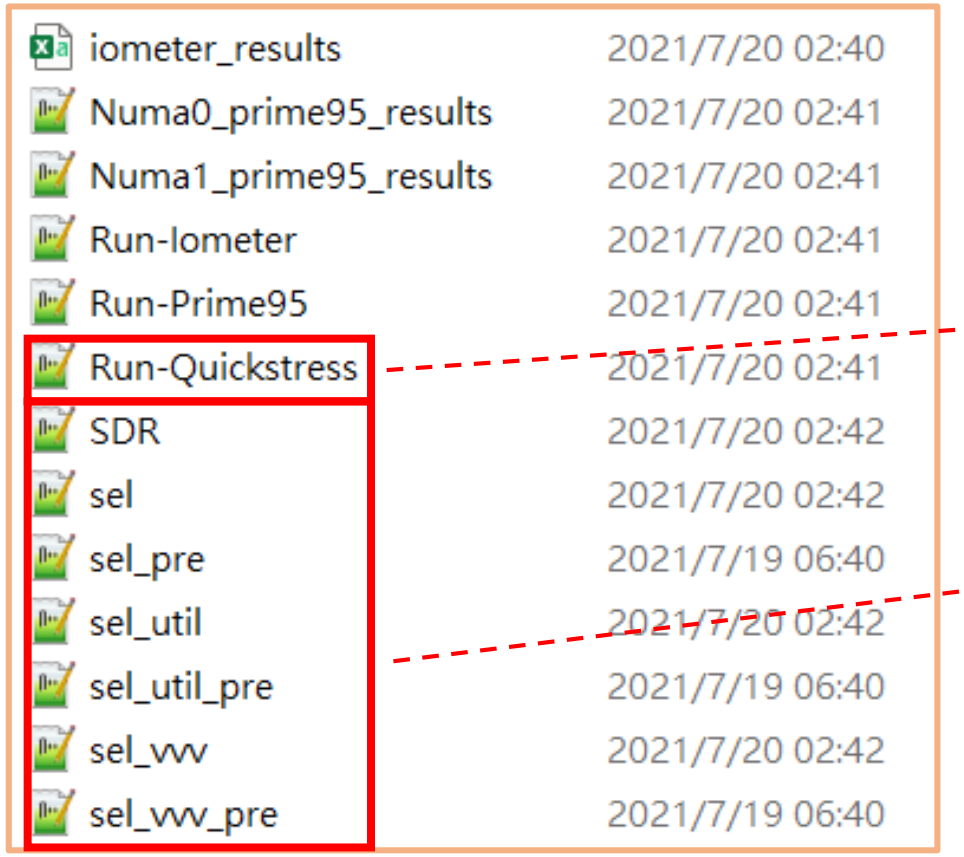
Reading and checking
The results after a test

Understanding the Test Result

- **A typical test will generate a lot of logs. Some of it will print out the test result for a quick look. Some of it are just raw data for further check if its needed.**
- **BAT logs will be placed in**
  - **C:\WcsTest\Result\Run-Quickstress\$TestTime$ [Test Summary]**
  - C:\WcsTest\Results\C2080_DV_BASELINE\Post-Test\Check-WcsError_$TestTime$
  - C:\WcsTest\Results\C2080_DV_BASELINE\Post-Test\Get-WcsConfig_$TestTime$
  - C:\WcsTest\Results\C2080_DV_BASELINE\Post-Test\Log-Msinfo32_$TestTime$
- **RDT logs will be placed in**
  - **C:\ToolkitResuls\Run-Quickstress\$TestTime$ [Test Summary]**
- **We will focus on how to judge the result from test summaries, but all logs need be saved after test so that we can find out more details.**

Understanding the Test Result

- **Files created after a test**



Test Summary
1) Test time and duration
2) Test item and results

SDR (Sensor Data Reading) / SEL (System Event Logs)
1) Ipmitool logs (before test and during tests)
2) Ipmiutil logs (before test and during tests)

Understanding the Test Result

- **Test Summary**
  - This file shows the test summary, including commands and test results

```
[6/18/2021 7:42:18 AM]    [Run-QuickStress called]
    + Called At line:1 char:37
    + Called As '. ..\WcsTest\Scripts\wcsScripts.ps1;Run-Quickstress -TimeinMin 720 -Nonet -Nofpga'
    + Called With Inputs: TimeInMin='720'  NoNetwork='True'  NoFpga='True'


[6/18/2021 7:42:18 AM]     Starting Run-IOmeter -Time 720 -LogDirectory C:\WcsTest\Results\Run-QuickStress\2021.06.18_07.42.18 -Full:False -NoWait

[6/18/2021 7:42:18 AM]     Starting Prime95 -Time 720 -LogDirectory C:\WcsTest\Results\Run-QuickStress\2021.06.18_07.42.18 -NoWait

[6/18/2021 7:42:22 AM]     Waiting 720 minutes for stress to complete

[6/18/2021 7:42:52 PM]    [Verify-QuickStress called]
    + Called At C:\WcsTest\Scripts\Library\StressLibrary.ps1:5689 char:24
    + Called As '$ErrorCount += Verify-QuickStress -LogDirectory $LogDirectory -NoDisks:$NoDisks -DiskSpeed:$DiskSpeed  -RunAsChild'
    + Called With Inputs: LogDirectory='C:\WcsTest\Results\Run-QuickStress\2021.06.18_07.42.18'  NoDisks='False'  DiskSpeed='False'  RunAsChild='True'


[6/18/2021 7:42:53 PM]     Prime95 passed

[6/18/2021 7:42:53 PM]     IOmeter passed

[6/18/2021 7:42:53 PM]    TEST PASSED!! (SIGNATURE) TEST PASSED!!
```

Understanding the Test Result

- **System Event Logs (SEL)**
  - Based on the commands we used, SEL can be summary-type and detailed-type.
  - A typical summary-type SEL looks like below picture. This log saves system event automatically, what we are looking for in this log is unusual messages like error/critical/fatal/fault/fail/incorrect.

```
 1 | 06/18/2021 | 07:41:41 | Event Logging Disabled SEL | Log area reset/cleared | Asserted
 2 | 06/18/2021 | 07:42:05 | Temperature Temp_CPU1 | Upper Non-critical going high | Reading 101 > Threshold 101 degrees C
 3 | 06/18/2021 | 07:42:05 | Fan PWM_1 |  | Asserted
 4 | 06/18/2021 | 07:42:05 | OEM record d0 | 9c9c00 | 000000000000
 5 | 06/18/2021 | 07:42:06 | OEM record d0 | 9c9c00 | 00000000ffff
 6 | 06/18/2021 | 07:42:06 | Temperature Temp_CPU1 | Upper Non-critical going high | Reading 93 > Threshold 101 degrees C
 7 | 06/18/2021 | 07:42:28 | Temperature Temp_CPU0 | Upper Non-critical going high | Reading 103 > Threshold 101 degrees C
 8 | 06/18/2021 | 07:42:28 | Fan PWM_1 |  | Asserted
 9 | 06/18/2021 | 07:42:28 | OEM record d0 | 9c9c00 | 000000000000
 a | 06/18/2021 | 07:42:29 | OEM record d0 | 9c9c00 | 00000000ffff
 b | 06/18/2021 | 07:42:33 | Temperature Temp_CPU1 | Upper Non-critical going high | Reading 101 > Threshold 101 degrees C
 c | 06/18/2021 | 07:42:33 | Fan PWM_1 |  | Asserted
 d | 06/18/2021 | 07:42:33 | OEM record d0 | 9c9c00 | 000000000000
 e | 06/18/2021 | 07:42:34 | OEM record d0 | 9c9c00 | 00000000ffff
 f | 06/18/2021 | 08:12:46 | Processor CPU0 Therm Stat |
10 | 06/18/2021 | 08:12:46 | Processor CPU0 Therm Stat |
11 | 06/18/2021 | 08:12:46 | Processor CPU0 Therm Stat |
12 | 06/18/2021 | 08:12:47 | Processor CPU0 Therm Stat |
13 | 06/18/2021 | 08:14:56 | Processor CPU0 Therm Stat |
14 | 06/18/2021 | 08:14:57 | Processor CPU0 Therm Stat |
```

Understanding the Test Result

- **Sensor Data Reading (SDR)**
  - SDR logs record every reading data from ipmitool. We usually use these log for (1) monitoring temperature, voltage and fan speed (2) checking timpstamp after unusual value showed up.

```
TIME 05-12-2021_23:50
PWM_1             | 100 unspecified   | ok
Energy_Storage    | no reading        | ns
NVDIMM            | 0x00              | ok
SM_ALERT          | 0x00              | ok
System Reconfig   | 0x00              | ok
HSC0 Input Power  | 1170 Watts        | ok
HSC0 Input Volt   | 12.10 Volts       | ok
HSC0 Output Curr  | 96 Amps           | ok
Temp_HSC0         | 62 degrees C      | ok
FPGA_S2_HSC_TMP   | no reading        | ns
FPGA_S2_DIE_TMP   | no reading        | ns
FPGA_S2_AMB_TMP   | no reading        | ns
FPGA_S2_NIC_TMP   | no reading        | ns
FPGA_S2_POWER     | no reading        | ns
CPU_ERROR         | 0x00              | ok
Temp_M.2_High     | 45 degrees C      | ok
FPGA_S1_HSC_TMP   | 51 degrees C      | ok
FPGA_S1_DIE_TMP   | 69 degrees C      | ok
FPGA_S1_AMB_TMP   | 56 degrees C      | ok
FPGA_S1_NIC_TMP   | no reading        | ns
FPGA_S1_POWER     | 80 Watts          | ok
FPGA_S3_HSC_TMP   | no reading        | ns
FPGA_S3_DIE_TMP   | no reading        | ns
FPGA_S3_AMB_TMP   | no reading        | ns
FPGA_S3_NIC_TMP   | no reading        | ns
FPGA_S3_POWER     | no reading        | ns
S1_NIC_TMP        | no reading        | ns
S2_NIC_TMP        | 59 degrees C      | ok
S3_NIC_TMP        | no reading        | ns
FPGA_CP_CORE_TMP  | 67 degrees C      | ok
FPGA_CP_AMB       | 0 degrees C       | ok
```

```
CP Health         | 0x00              | ok
Fan_PSU_1         | 19982 RPM         | ok
Fan_PSU_2         | 19879 RPM         | ok
Fan_PSU_3         | 19879 RPM         | ok
CPU0_VccIO        | 1 Volts           | ok
CPU1_VccIO        | 0.99 Volts        | ok
PSU_PWM_1         | 100 unspecified   | ok
BAT_Status_Event  | 0x00              | ok
SEL               | 0x04              | ok
Temp_PSU_HOTSPOT  | 68 degrees C      | ok
Temp_PSU_INLET    | 52 degrees C      | ok
Temp_PSU_OUTLET   | 58 degrees C      | ok
Temp_NIC_MAX      | 60 degrees C      | ok
Watchdog          | 0x00              | ok
CPU0_VccInPIN     | 245 Watts         | ok
CPU1_VccInPIN     | 235 Watts         | ok
CPU0_VccInPOUT    | 225 Watts         | ok
CPU1_VccInPOUT    | 240 Watts         | ok
Temp_CPU0_VR      | 84 degrees C      | ok
Temp_CPU1_VR      | 78 degrees C      | ok
CPU0_PROC_HOT     | 0x00              | ok
CPU1_PROC_HOT     | 0x00              | ok
PVCCP_CPU0_VRHOT  | 0x00              | ok
PVCCP_CPU1_VRHOT  | 0x00              | ok
DIMM0_ABCD_VRHOT  | 0x00              | ok
Temp_CPU0         | 79 degrees C      | ok
Temp_PCH          | 49 degrees C      | ok
Temp_Inlet        | 42 degrees C      | cr
Temp_Outlet       | 52 degrees C      | ok
Temp_CPU1         | 80 degrees C      | ok
Temp_DIMM_A       | 56 degrees C      | ok
Temp_DIMM_B       | 55 degrees C      | ok
```

```
Temp_DIMM_C       | 52 degrees C      | ok
Temp_DIMM_D       | 54 degrees C      | ok
Temp_DIMM_E       | 51 degrees C      | ok
Temp_DIMM_F       | 50 degrees C      | ok
Temp_DIMM_J       | 53 degrees C      | ok
Temp_DIMM_K       | 52 degrees C      | ok
Temp_DIMM_L       | 53 degrees C      | ok
Temp_DIMM_M       | 53 degrees C      | ok
Temp_DIMM_N       | 54 degrees C      | ok
Temp_DIMM_P       | 54 degrees C      | ok
DIMM0_EFGH_VRHOT  | 0x00              | ok
CPU0_MEM_HOT      | 0x00              | ok
CPU1_MEM_HOT      | 0x00              | ok
SSB ThermalTrip   | 0x00              | ok
P3V3              | 3.27 Volts        | ok
P5V               | 4.91 Volts        | ok
P0VDDQ_ABCD       | 1.22 Volts        | ok
P0VDDQ_EFGH       | 1.22 Volts        | ok
Temp_DIMM_MAX     | 56 degrees C      | ok
Temp_DIMM_G       | 55 degrees C      | ok
Temp_DIMM_H       | 53 degrees C      | ok
P0_VccIN          | 1.75 Volts        | ok
P1_VccIN          | 1.74 Volts        | ok
P12V_AUX          | 11.95 Volts       | ok
P1VDDQ_ABCD       | 1.22 Volts        | ok
P1VDDQ_EFGH       | 1.22 Volts        | ok
CPU0_ERR          | 0x00              | ok
CPU1_ERR          | 0x00              | ok
Temp_E1.S_High    | no reading        | ns
P5V_AUX           | 4.91 Volts        | ok
P1V05_STBY_PCH    | 1.04 Volts        | ok
Temp_DIMM_R       | 53 degrees C      | ok
```

```
Temp_DIMM_T       | 54 degrees C      | ok
DIMM1_JKLM_VRHOT  | 0x00              | ok
DIMM1_NPRT_VRHOT  | 0x00              | ok
PVNN_STBY_PCH     | 0.89 Volts        | ok
P3V3_STBY         | 3.31 Volts        | ok
P0VDDQ_ABCD_Curr  | 24 Amps           | ok
P0VDDQ_EFGH_Curr  | 23.20 Amps        | ok
P1VDDQ_ABCD_Curr  | 23.20 Amps        | ok
P1VDDQ_EFGH_Curr  | 23.20 Amps        | ok
P0_VccANA         | 1 Volts           | ok
P1_VccANA         | 0.99 Volts        | ok
P0_1V8S           | 1.79 Volts        | ok
P1_1V8S           | 1.79 Volts        | ok
P3V_BAT           | 3.13 Volts        | ok
P0_VccSA          | 0.76 Volts        | ok
P1_VccSA          | 0.78 Volts        | ok
Subsystem_Health  | 0x00              | ok
BMC Health        | 0x00              | ok
Fan_0_inlet       | 28840 RPM         | ok
Fan_0_outlet      | 24640 RPM         | ok
Fan_1_inlet       | 28980 RPM         | ok
Fan_1_outlet      | 24640 RPM         | ok
Fan_2_inlet       | 28840 RPM         | ok
Fan_2_outlet      | 24500 RPM         | ok
Fan_3_inlet       | 28980 RPM         | ok
Fan_3_outlet      | 24500 RPM         | ok
Fan_4_inlet       | 28980 RPM         | ok
Fan_4_outlet      | 24640 RPM         | ok
Fan_5_inlet       | 28980 RPM         | ok
Fan_5_outlet      | 24780 RPM         | ok
System Event      | 0x00              | ok
HyperV-Watchd     | 0x00              | ok
```

Appendix

- **Where can I find the PXE Server setup guide?**
  - Download it in GitHub: SetupPXE (Password: TeamMonica)
- **Where can I find the Switch setup guide?**
  - Download it in GitHub: C2080_INTRANET_QUICK_SETTING_V3 (Password: TeamMonica)
- **Where can I find the complete Test Plan for BAT and RDT?**
  - Download it in GitHub: C2080_PV_BAT_RDT_Testplan.zip (Password: TeamMonica)
- **Where can I find the Cross Table and Firmware Recipe?**
  - Download it in GitHub: C2080_PV_CT_FR.zip (Password: TeamMonica)
- **Where can I find the command that used for Rack Manager?**
  - Download it in GitHub: Project Olympus - Software CLI API Specification (Password: TeamMonica)
- **Where can I find the Test Scripts for reference?**
  - Download it in GitHub: Scripts (Password: TeamMonica)

Thanks!